
Neuro-Symbolic Integration of Hopfield Neural Network for Optimal Maximum Random k Satisfiability (MaxrkSAT) Representation

Hamza Abubakar^{1,*}, Sagir Abdu Masanawa² and Surajo Yusuf³

¹*School of Mathematical Sciences, Universiti Sains Malaysia (USM), Pulau Pinang, Malaysia*

²*Department of Mathematics, Federal University Dutsin-Ma (FUD), Katsina State, Nigeria*

³*Department of Mathematics, Isa Kaita College of Education, Dutsin-Ma, Katsina, Nigeria*

E-mail: zeeham4u2c@yahoo.com; amsagir@yahoo.com; surajoyusuf35@gmail.com

**Corresponding Author*

Received 24 February 2020; Accepted 22 June 2020;
Publication 28 October 2020

Abstract

Boolean satisfiability logical representation is a programming paradigm that has its foundations in mathematical logic. It has been classified as an NP-complete problem that difficult practical combinatorial optimization and search problems can be easily converted into it. Random Maximum k Satisfiability (MAX-RkSAT) composed of the most consistent mapping in a Boolean formula that generates a maximum number of random satisfied clauses. Many optimization and search problems can be easily expressed by mapping the problem into a Hopfield neural network (HNN) to minimize the optimal configuration of the corresponding Lyapunov energy function. In this paper, a hybrid computational model has been proposed that incorporates the Random Maximum k Satisfiability (MAX-RkSAT) into the Hopfield neural network (HNN) for optimal Random Maximum k Satisfiability representation (HNN-MAX-RkSAT). Hopfield neural network learning will be integrated

Journal of Reliability and Statistical Studies, Vol. 13, Issue 1 (2020), 199–220.

doi: 10.13052/jrss1550-4646.13110

© 2020 River Publishers

with the random maximum satisfiability to enhance the correct neural state of the network model representation. The computer simulation using C++ has been used to demonstrate the ability of MAX-RkSAT to be embedded optimally in Hopfield neural network to serve as Neuro-symbolic integration. The performance of the proposed hybrid HNN-MAXRkSAT model has been explored and compared with the existing model. The proposed HNN-MAXRkSAT demonstrates good agreement with the existing models measured in terms of Global minimum Ratio (Gm), Hamming Distance (HD), Mean Absolute Error (MAE) and network computation Time CPU time). The proposed framework explored that MAX-RkSAT can be optimally represented in HNN and subsequently provides an additional platform for neural-symbolic integration, representing the various types of satisfiability logic.

Keywords: Artificial neural networks, hopfield neural networks, wan abdulah method, boolean Satisfiability, random maximum ksatisfiability.

1 Introduction

Artificial Neural Networks (ANN) are considered as abstract computational models aimed at emulating the computational functional capacity of the biological neural network based on the brain dynamics computational model (Buscema et al., 2018). ANN teach the machine to perform tasks instead of training the software system to perform tasks. It is a pragmatic model that can quickly and precisely find patterns in the data. Core advantages of neural networks include the potential to understand complex or non-linear input-output interrelationships, applying serial training procedures and responses to a given data and provides a less technical approach to computation (Graves et al., 2016).

Hopfield neural networks (HNN) are a particular type of neural networks (NN) models that can store certain experiences, memories or patterns in a fashion similar to the human brain that the complete pattern can be retrieved if only partially or noisy information is presented to the networks. The dynamic behaviour of the HNN energy function has potential application capable of finding solutions to a difficult optimization problem. The HNN model is an underlying framework capable of processing such memories in a similar way to the nervous system brought major progress in the area of computational modelling and optimization with their ability to resolve complex real-world mathematical application (Abiodun et al., 2018).

The momentous breakthrough in ANN is Neuro-symbolic integration. Neural-symbolic architecture is AI systems that can perform any type of symbolic mechanisms within the artificial neural network framework. ANN has been regarded as connectionist systems. It draws the attention of various researchers in computational intelligence communities due to its ability to analyze and interpret complex non-linear phenomena including some mathematical and graphical models. (Kowalski, 1979) developed the main concept of mathematical and computational of informal logical representation as a programming language for the interpretation and analysis of a given problem. Abdullah (1992) was the pioneer in the field of by utilizing minimization capacity Hopfield neural network for logic programming representation and subsequently proposed a learning method to compute the optimal synaptic weight of HNN. Sathasivam, (2010) utilized and proved the success of the Wan Abdullah learning method to calculate the Horn logic programming synaptic weights in HNN. Sathasivam (2012) proposed the notion of a stochastic method in carrying out logic programming in HNN. Hamadneh et al., (2012) proposed an idea for representing logic programming incorporated into RBFNN as single operator logic. Velavan et al., (2015) proposed a flexible merger between logic programming and HNN in Mean Field Theory algorithm. Kasihmuddin et al. (2017) proposed a new searching technique that incorporates k SAT logical rule in the HNN. Alzaeemi et al., (2017) proposed the idea of incorporating logic programming in Kernel HNN. Kasihmuddin (2017) developed 2SAT logic representation incorporated with HNN. Mansor et al. (2017) has successfully upgraded the of 2SAT logic programming to 3SAT logic. The Max k SAT logic representation received attention from (Kasihmuddin et al. 2018). (Sathasivam et al., 2020) proposed the incorporation of R k SAT in HNN. However, in term of MAXR k SAT, there is no effort that combined the advantages of the non-systematical and practical application of R k SAT and maximum satisfiability of MAX k SAT logic program in HNN model. Therefore, we propose a new hybrid computational model by mapping MAXR k SAT in HNN in attaining better accuracy, sensitivity and robustness of higher-order networks. The contributions of the present study include; (1) To integrate the new logical representation, namely MAXR k SAT; (2) Mapping a MAXR k SAT logical rule into the discrete Hopfield neural network (HNN-MAXR k SAT-ES); (3) To explore the feasibility of HNN for optimal MAXR k SAT logical representation and measure its performance in term of accuracy with the existing models for HNN; (4) To establish a comprehensive comparison of the HNN-MAXR k SAT-ES with the existing HNN-KMAX k SAT-ES (Kasihmuddin et al, 2017), KHNN-R k SAT-ES

(Sathasivam et al., 2020) and KHNN- k SAT-ES (Alzaeemi et al., 2017) models in the literature. By developing an integrated ANN working model, the proposed hybrid computational model will be beneficial by proving an alternative method of computation in finding the optimal representation of various mathematical optimizations and search problem.

This paper proceeds as follows: Section 2 presented the proposed Random Maximum k Satisfiability (MAXR k SAT). In Section 3 presented the mapping of Random Maximum k Satisfiability in Hopfield neural network is reported. The Author presented Wan Abdullahi method for synaptic weight computation in Section 4.s Section 6 presented a new learning rule (NLR) and relaxation method for learning in HNN model. Section 6 covers the HNN-MAXR k SAT simulation design & experimental setup. In Section 7 and 8 reported the model performance measure and the experimental Results respectively. Finally, the remaining sections reported the results, discussion and conclusion for future direction.

2 The Proposed Random Maximum k Satisfiability (MAXR k SAT)

Random Maximum k Satisfiability (MAXR k SAT) is a class of non-systematic Boolean satisfiability representation that composed a maximum number of random literals per clause to be negated with the probability of 0.5. The MAXR k SAT logic can be represented in CNF where each logical clause consists of a random number of Boolean variables connected by a logical operator. The standard structure of MAXR k SAT logical representation is restricted as compared to the ordinary k SAT(Yolcu and Póczos, 2019) and MAX k SAT logical representation. However, R k SAT part of our problem is not restricted (Sathasivam et al., 2020). The general formulation for MAXR k SAT will be restricted obeying Equation (1);

$$Q_{MAXR2SAT} = \bigwedge_{i=0}^n Q_{MAXk2SAT} \bigwedge_{i=0}^m Q_{R2SAT} \quad (1)$$

where Q_{R2SAT} and $Q_{MAX2SAT}$ are defined in Equation (2) and Equation (3) respectively follows;

$$Q_{R2SAT} = \bigwedge_{i=0}^n C_i^{(2)} \bigwedge_{i=0}^m C_i^{(1)} \quad (2)$$

$$Q_{MAX2SAT} = \bigwedge_{i=0}^n \lambda_i^{(2)} \bigwedge_{i=0}^m \beta_i^{(2)} \quad (3)$$

where $\forall m, n \in [1, 2, \dots, N]$, $n > 0$ and $m > 0$. The clauses in Q_{R2SAT} is presented in Equation (5) and $Q_{MAX2SAT}$ are defined in Equation (6) and Equation (7) respectively as follows;

$$C_i^{(k)} = \begin{cases} (I_i \vee J_i), & k = 2 \\ L_i & k = 1 \end{cases} \quad (4)$$

$$\lambda_i^{(2)} = (\lambda_1 \vee \lambda_2) \wedge (\neg\lambda_1 \vee \lambda_2) \wedge (\lambda_1 \vee \neg\lambda_2) \wedge (\neg\lambda_1 \vee \neg\lambda_2) \quad (5)$$

$$\beta_i^{(2)} = (\beta_1 \vee \beta_2) \quad (6)$$

where $C_i^{(1)}$ and $C_i^{(2)}$ designated as the first and second-order logical clauses respectively in Q_{R2SAT} . $B_i^{(2)}$ and $V_i^{(2)}$ designated as the second-order clause in $Q_{MAX2SAT}$. In this work, F_α used to represent a Boolean formula in CNF where logical clauses are chosen uniformly, independently and without any replacement from $2^\alpha \binom{m+n}{\kappa}$ non-trivial clause of length α . I_i exists

in the $C_i^{(k)}$, if the $C_i^{(k)}$ contains either I_i or its negation ($\neg I_i$) and the mapping of $g(F_\alpha) \rightarrow [-1, 1]$ defined as a logical interpretation of Boolean formula. Sathasivam et al. (2020) [15], Alzaeemi et al. (2017) described that any Boolean value for the mapping of Satisfiability representation can be expressed as 1 or -1 for TRUE or FALSIFICATION respectively. Theoretically from Equation (1), $Q_{MAXR2SAT}$ for $k \leq 2$ can be mathematically presented as follows;

$$Q_{MAXR2SAT} = \underbrace{\left\{ (\lambda_1 \vee \lambda_2) \wedge (\lambda_1 \vee \lambda_2) \wedge (\lambda_1 \vee \lambda_2) \wedge (\neg\lambda_1 \vee \lambda_2) \wedge (\beta_1 \vee \beta_2) \right\}}_{MAX-2SAT} \wedge \underbrace{\left\{ (I_1 \vee \neg J_1) \wedge (I_2 \vee \neg J_2) \wedge \neg L_1 \right\}}_{RANDAM-2SAT} \quad (7)$$

According to Equation (6), $Q_{MAXRkSAT}$ comprises of $C_1^{(2)} = (I_1 \vee \neg J_1)$, $C_2^{(2)} = (\neg I_2 \vee J_2)$, $C_1^{(1)} = \neg L_1$, $\lambda_i^{(2)} = (\lambda_1 \vee \lambda_2) \wedge (\neg\lambda_1 \vee \lambda_2) \wedge (\lambda_1 \vee \neg\lambda_2) \wedge (\neg\lambda_1 \vee \neg\lambda_2)$ and $\beta_i^{(2)} = (\beta_1 \vee \beta_2)$. Therefore, the result of Equation (7) is reduced to $Q_{MAXRkSAT} = -1$ (not satisfiable). Hence, Equation (7) is considered as one of the constrained optimizations and search problems that can be found maximization problem. Kasihmuddin et al (2018) and Mansor et al. (2017) observed that MAXkSAT is not fully satisfiable, it is therefore considered as a constrained optimization and search problem that can be carried out on the HNN model for optimal representation.

3 Mapping of Random Maximum t Satisfiability in Hopfield Neural Network

Hopfield neural networks (HNN) are a computational method of biological influence that can be applied in various combinatorial and search problems. Their advantage over more traditional optimization techniques is their ability to use strong computational power in discrete components and the inherent parallelism of the network. HNNs are recurrent types of ANN-based on content-addressable memory with binary threshold nodes that should yield a minimum local. HNN basic architecture and structure consists of discrete interconnected bipolar represent a neural computational model through an auto-associative framework behaviour with a strictly symmetrical weight matrix between the neurons without any self-loop and hidden neurons. Given an initial state vector variable $S_i (i = 1, 2, 3, \dots, n)$ that is input to the HNN model. The HNN will converge to the state of equilibrium corresponding to the minimum value $H_{Q_{MAXRkSAT}}$. The HNN minimizes the energy function due to its apparent resemblance to the physical revolving system in statistical mechanics (Barra et al. 2018). The neuron's state in HNN is considered bipolar, $D_i \in [-1, 1]$ comply with the dynamics by obeying the general asynchronous updating rule as follows:

$$D_i(t+1) = \begin{cases} 1, & \text{if } \sum_j M_{ij} D_j(t) + \Psi \\ -1, & \text{otherwise} \end{cases} \quad (8)$$

where M_{ij} designated the HNN strength vector that maintains the link between the connections neuron j to another neuron i with pre-determined bias Ψ . The MAXRkSAT logical representation can be mapped into HNN by assigning each vector variable with neurons D_i with the represented cost function. Hence, the cost function $E_{Q_{MAXRkSAT}}$, that man the combinations of HNN and MAXRkSAT logic is represented as follows:

$$E_{Q_{MAXRkSAT}} = \sum_{i=1}^{NC} \prod_{j=1}^m T_{ij} \quad (9)$$

where NC and m represent the number of logical clauses and the number variables $Q_{MAXRkSAT}$ logic respectively. The inconsistency of a logical

clause in $Q_{MAXRkSAT}$ is given as follows:

$$T_{ij} = \begin{cases} \frac{1}{2}(1 - D_x), & \text{if } \neg x \\ \frac{1}{2}(1 + D_x), & \text{otherwise} \end{cases} \quad (10)$$

MAXRkSAT logic in HNN can be updated obeying the following:

$$h_i(t) = \sum_{j=1, i \neq j}^N M_{ij}^{(2)} D_j(t) + M_i^{(1)} \quad (11)$$

$$D_i(t+1) = \begin{cases} 1, & \sum_{j=1, i \neq j}^N M_{ij}^{(2)} D_j(t) + M_i^{(1)} \geq 0 \\ -1, & \sum_{j=1, i \neq j}^N M_{ij}^{(2)} D_j(t) + M_i^{(1)} < 0 \end{cases} \quad (12)$$

where $M_{ij}^{(2)}$ and $M_i^{(1)}$ represent the second and the first order synaptic connection of HNN integrated with MAXRkSAT logic. Equation (11) to Equation (12) are vital stages to ensure that neurons state in HNN converges to an optimal configuration corresponding to MAXRkSAT logical representation. To assess the quality of the state pattern recovered in the network, the Lyapunov energy function, $H_{Q_{MAXRkSAT}}$ described in Equation (13) was applied as follows.

$$H_{Q_{MAXRkSAT}} = -\frac{1}{2} \sum_{i=1, i \neq j}^N \sum_{j=1, i \neq j}^N M_{ij}^{(2)} D_i D_j - \sum_{i=1}^N M_i^{(1)} D_i \quad (13)$$

One of the properties possessed by Lyapunov energy function in Equation (13) is that the energy viewed from the $Q_{MAXRkSAT}$ always decreases monotonically with the network configuration. The value of $H_{Q_{MAXRkSAT}}$ in determining the value of network energy corresponding to the absolute final energy of the network consumed $H_{Q_{MAXRkSAT}}^{\min}$. Therefore, the overall quality of the final state of the network can be properly obeyed the given condition (Sathasivam, 2012).

$$|H_{Q_{MAXRkSAT}} - H_{Q_{MAXRkSAT}}^{\min}| \leq \xi \quad (14)$$

where ξ defined by any pre-determined tolerance value. According to Sathasivan (2020), 0.001 is appropriate. Note that if the embedded logic clause in HNN does not meet the condition in Equation (14), then the final state pattern obtained is assumed to be stuck in a local minimum solution (ie wrong pattern). Worth mentioning that, $M_{ij}^{(2)}$ and $M_i^{(1)}$ can be effectively computed by using Wan Abdullah learning method (Abdullah, 1992) which is equivalent to Hebbian Learning rules (Gerstner and Kistler, 2002). In this paper, the authors donated the mapping of MAXRkSAT logic in HNN is as HNN-MAXRkSAT logical representation.

4 Wan Abdullahi Method for Synaptic Weight Computation

MAXRkSAT logic can be represented as one of the constrained optimization and search problems that are being carried out on HNN. Wan Abdullah logic learning became the pioneer in computing the synaptic weight of current neural network like HNN based on logical inconsistencies (Sathasivam et al., 2020). The cost function of the problem that corresponds to MAXRkSAT logical is represented in the minimized form of logical inconsistencies in $Q_{MAXRkSAT}$ defined as follows.

$$\min_{i \in (0, \infty), Q_{MAXR2SAT}=1} \neg Q_{MAXR2SAT} \quad (15)$$

Equation (14) is considered as one of the constrained optimization/decision problems that can be found maximization problem. Finding inconsistencies of Equation (7) can be represented by its negation as follows;

$$\begin{aligned} \neg Q_{MAXR2SAT} = & (\lambda_1 \vee \lambda_2) \wedge (\neg \lambda_1 \vee \lambda_2) \wedge (\lambda_1 \vee \neg \lambda_2) \wedge (\neg \lambda_1 \vee \neg \lambda_2) \\ & \wedge (\beta_1 \vee \beta_2) \wedge (\neg I_1 \vee J_1) \wedge (I_2 \vee \neg J_2) \wedge L_1 \end{aligned} \quad (16)$$

The cost function for (16) is defined as follows;

$$\begin{aligned} E_{Q_{MAXR2SAT}} = & \frac{1}{2}(1 - C_{\lambda_1})\frac{1}{2}(1 - C_{\lambda_2}) + \frac{1}{2}(1 + C_{\lambda_1})\frac{1}{2}(1 - C_{\lambda_2}) \\ & + \frac{1}{2}(1 - C_{\lambda_1})\frac{1}{2}(1 + C_{\lambda_2}) + \frac{1}{2}(1 - C_{\lambda_1})\frac{1}{2}(1 - C_{\lambda_2}) \\ & + \frac{1}{2}(1 - C_{\beta_1})\frac{1}{2}(1 - C_{\beta_2}) + \frac{1}{2}(1 - C_{I_1})\frac{1}{2}(1 + C_{J_1}) \\ & + \frac{1}{2}(1 - C_{I_2})\frac{1}{2}(1 + C_{J_2}) + \frac{1}{2}(1 + C_{L_1}) \end{aligned} \quad (17)$$

Table 1 Displays the clauses and corresponding synaptic weights

Synaptic weights	$(\neg\beta_1 \vee \neg\beta_2)$	$(\neg I_1 \vee J_1)$	$(\neg I_2 \vee J_2)$	L_1	HNN – MAXR2SAT
$C_{\beta_1} C_{\beta_2}$	$-\frac{1}{4}$	0	0	0	$-\frac{1}{4}$
C_{β_1}	$\frac{1}{4}$	0	0	0	$\frac{1}{4}$
C_{β_2}	$\frac{1}{4}$	0	0	0	$\frac{1}{4}$
$C_{I_1} C_{J_1}$	0	$\frac{1}{4}$	0	0	$\frac{1}{4}$
C_{I_1}	0	$-\frac{1}{4}$	0	0	$-\frac{1}{4}$
C_{J_1}	0	$\frac{1}{4}$	0	0	$\frac{1}{4}$
$C_{I_2} C_{J_2}$	0	0	$\frac{1}{4}$	0	$\frac{1}{4}$
C_{I_2}	0	0	$\frac{1}{4}$	0	$\frac{1}{4}$
C_{J_2}	0	0	$-\frac{1}{4}$	0	$-\frac{1}{4}$
C_{L_1}	0	0	0	$-\frac{1}{2}$	$-\frac{1}{2}$

The Appropriate weight matrix-vector of HNN-MAXR2SAT can be obtained by equating $E_{Q_{MAXR2SAT}}$ in Equation (17) with $H_{Q_{MAXRkSAT}}$ in Equation (13) and the result is displayed in Table 1.

The synaptic weight connections calculated and presented in Table 1 are sums of the inputs of individual logical clauses of HNN-MAXR2SAT to be stored CAM of HNN which will later be used in the retrieval phase. The synaptic weights are essential bits of information that are acquired after the training process. The training process in this experiment yields the optimum value of cost function that will determine the system's effective weights. The optimal global minimum energy estimation requires suitable mapping and adjusted synaptic weights. The optimal global minimum energy can be delineated at the start of the recovery process as the projected global minimum energy. The logic program of MAXR2SAT can be regarded as combinatorial optimization. This is done by minimizing the logical inconsistency of the Boolean formula. The general global minimum energy for the HNN-MAXR2SAT is given as in Equation (17). The satisfied interpretation such as $C_{\beta_1} = 1, C_{\beta_2} = 1, C_{I_1} = 1, C_{J_1} = -1, C_{I_2} = 1, C_{J_2} = -1, C_{L_1} = 1$ is substituted into Equation (17). We obtained the optimal globally minimum energy in Equation (18) since $E_{HNN-MAXRkSAT}^{min}$ is not satisfiable we obtained the optimal satisfiability as follows.

$$E_{MAXR3SAT}^{Optimum} = -\frac{3}{4} \quad (18)$$

$E_{MAXR3SAT}^{Optimum}$ will be used to separate the correctness of the neuron state produced by the network during the retrieval phase.

5 New Learning Rule (NLR)

The activation function is a complex model of Hopfield's neural network of logic programming. The activation mechanism used in neural-symbolic integration to turn the activation rate of a system (neuron) into an output signal (Sathasivam, 2015). Nonetheless, this trigger mechanism imposes too much focus on mild noise disturbance instead of the cost-related signals (Alzaemi et al., 2017) proposed a new activation function as follows:

$$f_{X_i} = \frac{\frac{1}{2} \left(1 + \tanh\left(\frac{\nu_{X_i}}{u_0}\right) \right)}{1 + \tanh\left(\frac{x_0}{u_0}\right)} \quad (19)$$

$$f_{X_i} = \frac{\tanh\left(\frac{x_0}{u_0}\right) + \frac{1}{2} \left(1 + \tanh\left(\frac{\nu_{X_i} - x_0}{u_0}\right) \right)}{1 + \tanh\left(\frac{x_0}{u_0}\right)} (\nu_{X_i} \geq 0) \quad (20)$$

where the f_{X_i} and ν_{X_i} represent the activation function and the initial states of the HNN respectively. x_0 define the threshold value f_{X_i} to become steeper, and u_0 use to compute the triggering function's steepness. It can handle this function with noise and do well if the system is large.

5.1 The Relaxation Method for Learning in HNN Model

The relaxation method is a useful stage for contextual information processing to reduce local uncertainty and maintain the global accuracy of the HNN model. It is essentially a parallel execution system that changes the confidence levels of the entities involved based on interrelated theories and measurements of confidence. The neural network, on the other hand, is a computational paradigm with massive parallel capacity for execution. Each neuron's output depends primarily on information from other neurons. Therefore, in the relaxation process and the neural network technique, there are some common properties. (Sathasivam, 2015; Kasihmuddin et al., 2019) proposed a mapping which allows HNN to perform the relaxation process. A downside of this is that the relaxation process can be carried out in real-time

because traditional analogue circuits can implement the HNN model. The neural network design can be easily adapted by this approach to solve the many problems that the relaxation mechanism has already solved. “HNN–MAXRkSAT relaxation” can be demarcated after the local field has been collected as a sequence of relaxing loops in the system. The network tends to create many local minima solutions lacking an adequate relaxation mechanism. The relaxation frequency function is to modify the relief speed so that higher quality solutions can be obtained. Because HNN–MAXRkSAT includes further clause restrictions, authors allow network to pass through relaxation stage to ensure that it stays calm in its final state. The HNN–MAXRkSAT relaxation obeys the following function.

$$\frac{dh_i^{new}}{dt} = \frac{Rdh_i}{dt} \quad (21)$$

where R refers to the relaxation rate, h_i^{new} refers to the updated local field and h_i defined as the local field value of the model measured based on HNN–MAXRkSAT. The relaxation frequency R theoretically reflects how quickly the model is relaxed. The R value is a modifiable parameter that can be empirically computed. The optimum relaxation rate is usually in the range of $2 \leq R \leq 4$. In this paper, we deployed $R = 4$ for all of the simulations. The selected relaxation rate complies with the work (Sathasivam, 2015).

6 HNN-MAXRkSAT Simulation Design and Experimental Setup

In this study, the MAXRkSAT logic to be incorporated into HNN to search for optimal MAXRkSAT logic representation. HNN model employed simulated data sets to implement MAXRkSAT logical clauses. HNN-MAX-RkSAT simulations were performed with Dev C++ release version 5.11 on Windows 8, Intel Core i3, 1.7 GHz 8 GB RAM processor. Initially, the program must randomize the neuron position. The primary objectives of this project were to seek an optimal model, which would represent the practical MAXRkSAT model. The algorithms Figure 1 displays the implementation of the HNN model within the system. The following algorithms illustrate how the proposed models are implemented in the program. Table 2 indicates the appropriate control parameters utilized during each HNN model implementation.

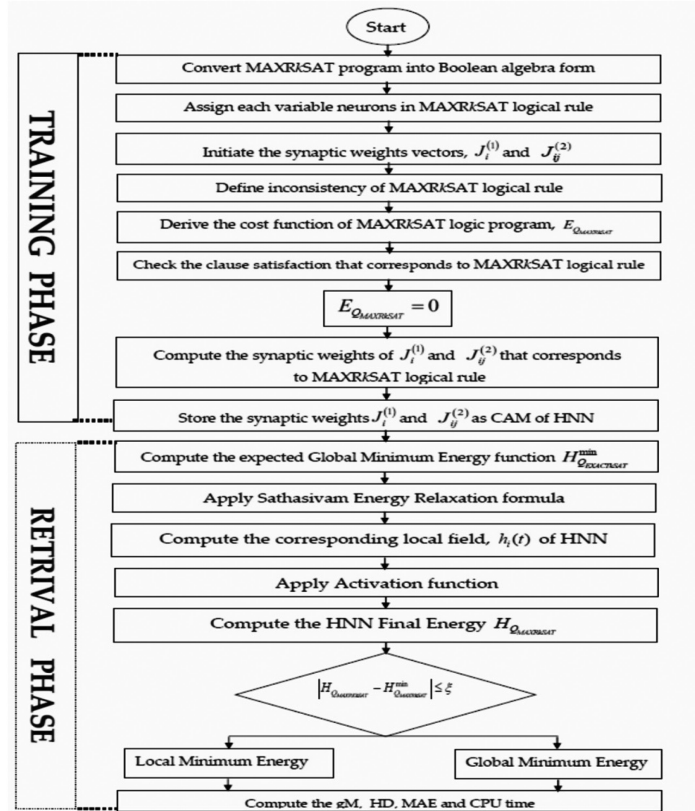


Figure 1 Flowchart for HNN-MAXRkSAT implementation procedure.

Table 2 List of parameters and their value used in HNN

Parameter	Parameter Value
Number of clauses (NC)	40
Neuron Combination (NN)	100
Tolerance Value (ξ)	0.001
Number of Learning (ϑ)	100
Selection Rate (α)	0.1
Number of trials (τ)	100
Relaxation time (R)	4

7 Model Performance Matrics

A total of four performance different matrix measure were employed to explore the performance of HNN-MAXRkSAT logical representation

compares with the existing model. The proposed HNN-MAXRkSAT model was assessed based on Global Minimum Ratio (gM), Hamming Distance (HD), Mean Square Error (MSE) and Computation time (CPU time). The performance matrices utilized in this study have been discussed as follows:

7.1 Global Minima Ratio (gM)

Global Minima Ratio (gM) is described as the proportion between the total minimum global energy and maximum runs. (Sathasivam, 2012; Kasihmuddin et al., 2019). Because the HNN model will generate 10,000 solutions per execution, this analysis will make searching for gM applicable. Each of the neurons measured $H_{Q_{MAXRkSAT}}$ in HNN will be sorted based on ξ . If the HNN is within the condition states in Equation (14), then the $H_{Q_{MAXRkSAT}}$ will be regarded as the minimum global energy. gM obeys the following equation.

$$gM = \frac{1}{\tau c} \sum_{i=1}^{NC} N_{H_{Q_{MAXRkSAT}}} \quad (22)$$

where τ and c represent the number of trial and the combination of neurons in MAXRkSAT respectively. The amount of the proposed model's global minimum energy. If the value of gM corresponds to 1 a particular model is rendered robust.

7.2 Proportion of Satisfied Clause (PSC)

Since MAX-RkSAT logical rule cannot be satisfied, the program will evaluate the proportion of clauses that are satisfied (PCS). The following equation will be used to calculate the proportion of the clause satisfied.

$$PSC = \frac{f_{MAXRkSAT}}{NC} \quad (23)$$

where $f_{MAXRkSAT}$ and NC denote the fitness of MAXRkSAT logical clause and a total number of clauses in HNN-MAXRkSAT model respectively. The efficiency of the MAX-RkSAT model will be measured based on an increasing amount of several neurons.

7.3 Hamming Distance (HD)

Many problems in information storage, retrieval and related fields depend on an accurate measurement of the distance or similarity between objects

most often represented as vectors. In this paper, HD determines the proximity of bits in the relaxation cycle between the stable state and the global state (Guoguang et al. 2008). We use the HD to compute the difference between the original pattern and the patterns retained, or between the output pattern and the patterns stored. The HD is a computation used to equate two binary patterns in the HNN, which is the number of bits of the two patterns varying. The distance to HD here is defined as follows;

$$HD = \sum_{i=1}^{NC} |s_i - s_i^\tau| \quad (24)$$

where s_i is a state of an initial state presented to the network or an output pattern generated, s_i^τ is the i th component of the τ th stored pattern that was presented. In our study, the τ th stored pattern, the HD of the network trend would be 5 or 40, whether the initial state or output pattern is the τ th pattern stored or its exactly reversed form respectively.

7.4 Mean Absolute Error (MAE)

Mean Absolute Error (MAE) is the one of the best performance metric for displaying the uniformly distributed error generated in a model (Chai et al. 2014). The estimation of the MAE takes the absolute value of the difference between the expected values and the real values. A good model with HNN-MAXRkSAT will have the lowest MAE value. The MAE equation is shown below:

$$MAE = \sum_{i=1}^{NC} \frac{1}{NC} |f_x - f_{\max}| \quad (25)$$

Where $f_{MAXRkSAT}$ and f_i donated the maximum fitness and fitness value observed respectively.

7.5 Computation Time (CPU Time)

Computation time (CPU Time) has been one of is an important metric or indicator for analyzing our model efficiency. It involves learning and extracting the total satisfied clauses through our proposed framework. The CPU time is represented at the time that a particular network has consumed to complete one execution. It is about the measure of effectiveness and stability of ANN models.

$$CPU _ Time = Learning _ Time + Recovery _ Time \quad (26)$$

8 Experimental Results and Discussion

Figures 2–6 show search performances between the proposed model and other existing models. It exhibits the behaviour of the errors observed in the searching process from $5 \leq NN \leq 40$. In general, based on the simulated results the HNN-MAXRkSAT logic manifests good agreement with the existing models in the literature.

Figure 2 displays the zM trend via HNN-MAXRkSAT in compares HNN-KMAXkSAT-ES (Kasihmuddin et al. 2018), (Sathasivam et al., 2020), KHNN-RkSAT-ES and KHNN-kSAT-ES(Alzaeemi et al. 2017) for optimal logical representation. Its efficiency can be calculated by testing the

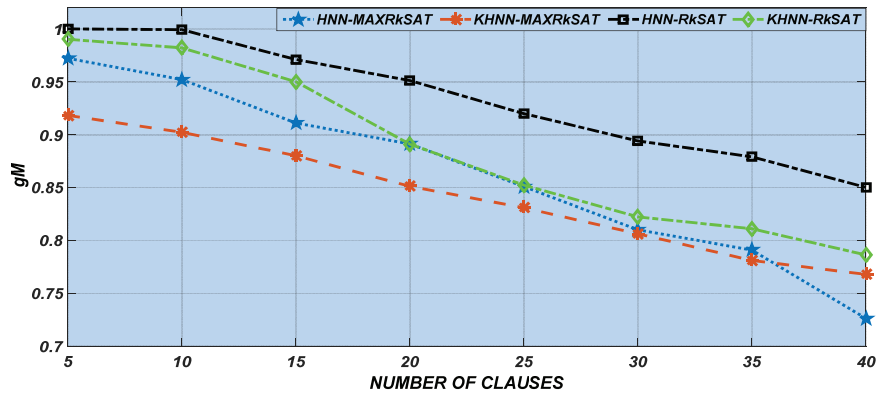


Figure 2 Global minimum ratio (gM) of various models.

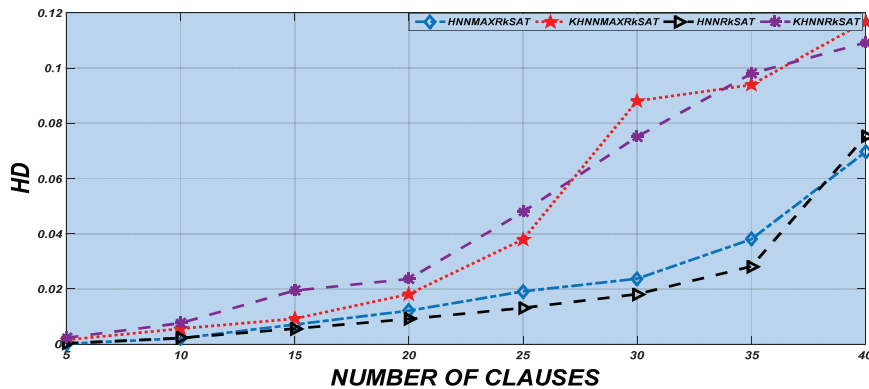


Figure 3 Hamming distance (HD) performance evaluation of various models.

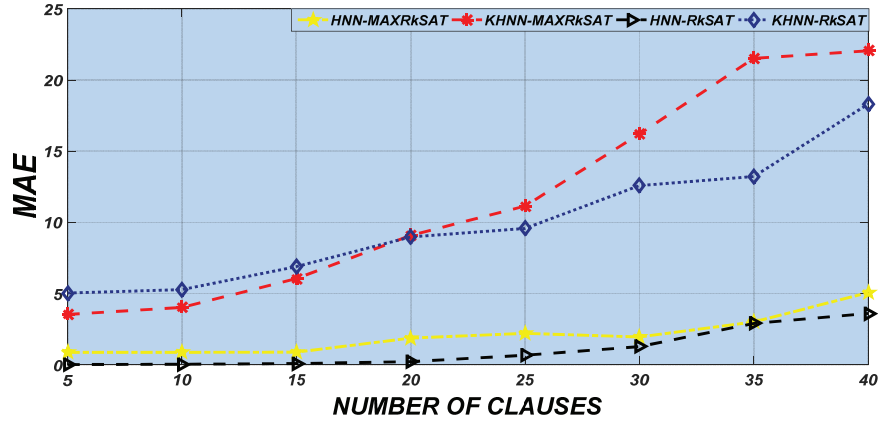


Figure 4 Mean absolute error (MAE) performance evaluation of various models.

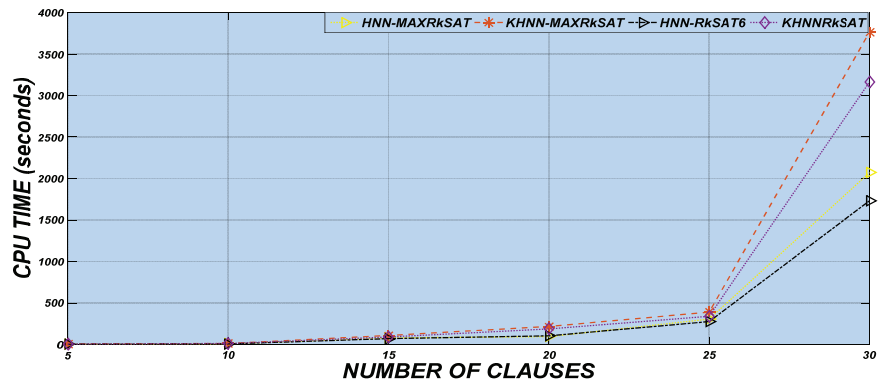


Figure 5 Computational time (CPU time) performance evaluation of various models.

consistency of the energy from $5 \leq NC \leq 40$. $gM = 0.8502$ is identified as 8502-bit strings of minimum global energy and 598-bit strings of minimum local energy. If the gM of the network approaches one, almost all neurons achieved the required final state during the recovery phase. An effective method of calming by Sathasivam relaxation system stabilizes the state of the neuron during the period of recovery. The stable neuron state generated by HNN-MAXRkSAT results on the convergence of the energy generated to global minimum energy. As the number of neurons rose (MAX-RkSAT maximum number increased), some of the collected neuron states may be stuck in a local minimum solution (suboptimal solution). All HNN models entail more computation time before the model can enter the relaxation phase to avoid MAX-RkSAT's inconsistencies. It can also stabilize the

neuronal system by crushing the neuronal collective output. This triggers the neural systems to converge to the global minimum. Owing to the random nature of the search for possible solutions, HNN-MAX-RkSAT agreed with the performance of the existing models in the literature as demonstrated in Figure 2.

Figure 3 shows the relation between the HD and the NC. It demonstrates the trend display by all HNN models in term of Hamming distance from $5 \leq NC \leq 40$. The Hamming distance in this experiment represents the proximity of the neuron's state between the learning state and retrieved state (Global solution) of the neurons when relaxing. The trend demonstrates a steadily improved in the performance of HNN-MAXRkSATES. The trend in which HNN-MAXRkSATES can remember the correct statements that led to the lower HD demonstrates similar behaviour with the existing model. It is noticeable that KHNN model accumulates higher rise in HD than HNN model. The exhaustive search algorithm, on the other hand, highlighted the process of trial and error during the fulfilment process of the rule. According to the random nature of the search space in searching the right states, HNN-MAX-RkSATES were able to accommodate $NC \geq 40$ when the complexity of the network increased. Unlike KHNNMAXkSATES that can only sustain $5 \leq NC \leq 40$ as observed in Figure 3. Similar behaviours were observed by the previous work as justified in (Kasihmuddin et al. 2017; Alzaeemi, et al. 2017; Sathasivam, et al. 2020). The major reason is due to the exhaustive search design, which raises the mathematical workload when looking for the right neuron states.

Figure 4 shows the relation between the MSE and the NC. It can be observed in Figure 4 that the learning error in terms of MAE increases massively as the neurons out weight from $NC \geq 10$. The comparison in the HNN model trend, based on MAE error measures. HNN-MAXRkSAT and HNNRkSAT display the lowest performance recording 5.07 (94.93% accuracy) and 3.61 (96.39% accuracy) respectively. KHNN-MAXRkSAT and KHNNRkSAT have the highest MAE accumulating closed in 22.05 (77.95% accuracy) and 18.31 (81.69% accuracy) respectively. All models display the good performance agreement of close to 95% at the initial stage when $NC = 5$ and 80% on the final stage when $NC = 40$. A rapid increase in MAE error is noticed in all models. The justification for this trend is that a higher as the complexity of NC raised (Peter et al. 2017). However, the gap between all is not significant. This indicates that HNN-MAXRkSAT agreed with other models, both in the short run and in the long-run trend.

Based on Figure 6, the computation time was displayed for all models, it is observed that all models under study start execution with a reasonable

amount of execution time. At $NC = 40$, CPU time could not be plotted as the execution time is too high (that is it exceed a threshold value). The figure displays CPU time the trend from $5 \leq NC \leq 30$. At $NC = 5$, HNN executes MAXRkSAT and RkSAT logic in 2.5 seconds and 1.5 seconds respectively, and KHNN executes MAXRkSAT and RkSAT in 6.8 seconds and 5.1 seconds respectively. At $NC = 40$, KHNN consume 12981 seconds to execute MAXRkSAT logic while HNN consumes 7871.8 seconds to execute RkSAT logical representation. Therefore, HNN requires less computational time compared to other models in executing RkSAT logic. However, all models maintain a similar pattern even as the number of neurons increases. The computational time for all model raised high. The Justification for this trend is that a higher as the number of clauses high. As it stands, HNNRkSATES requires a substantial amount of time to reach the optimal eligibility (Sathasivam, 2015; Kasihmuddin et al., 2019; Sathasivam et al., 2020). This problem can be reduced by employing the metaheuristics algorithm. The KHNN-MAXRkSAT computation time is higher than other existing models in the literature. During the training phase of HNN, the optimal searching technique is needed to drive the solution to optimal eligibility in an acceptable time range. The searching techniques will work extensively throughout the training phase for doing MAXRkSAT programming as the complexity of the neurons rises.

9 Conclusion

We have successfully developed a model that explored the feasibility of the Hopfield neural network (HNN) to be incorporated with MAX-RkSAT programming. A network by incorporating with Hopfield neural network in performing maximum random k-satisfiability logic programming (HNN-MAXRkSATES). The proposed model was compared with other existing models to measure its performance. The work, reported in this paper, revealed the efficient performance of HNN-MAXRkSAT model in terms of the global minimum ratio, Hamming distance, Mean Absolute Error and the computation time. According to the experimental results, the proposed HNN-MAXRkSAT gives us an acceptable result and also agreed with KHNN-MAXRkSAT, KHNNRkSAT and HNNRkSAT in all metric measures enjoyed in this study. The proposed framework provides a solid platform for evaluating various types of satisfiability problem.

The upcoming research in this regard will be focusing on the investigation to explore the other variants of the satisfiability representation problem such as minimum satisfiability, restricted Maximum Random satisfiability, weighted maximum random satisfiability and quantified maximum random

satisfiability. We will also propose HNN-MAXRkSAT to accommodate real-life data set, that, Hopfield neural network Random Maximum k Satisfiability Reverse Analysis (HNN-MAXRkSATRA). Additionally, robust metaheuristic techniques such as Genetic Algorithm (GA), Election Algorithm (EA) and swarm intelligence like Artificial bee colony (ABC) and Particle swarm (PSO) to reduce the complexity of the HNN-MAXRkSAT model during the training phase and other intelligence algorithms can be integrated with HNN-MAXRkSAT to accelerate the learning phase for optimal representation.

References

- Abdullah, W. A. T. W. (1992). Logic programming on a neural network, *International journal of intelligent systems*, 7(6), pp. 513–519.
- Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A. and Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey, *Heliyon*, 4(11), e00938.
- Alzaeemi, S. A. S., Sathasivam, S., Adebayo, S. A., Kasihmuddin, M. S. M. and Mansor, M. A. (2017). Kernel machine to doing logic programming in Hopfield network for solve non horn problem-3sat, *MOJ Applied Bionics and Biomechanics*, 1(1), pp. 1–6.
- Barra, A.; Beccaria, M. and Fachechi, A. A. (2018). New mechanical approach to handling generalized Hopfield neural networks, *Neural Network*, 106, pp. 205–222.
- Buscema, P. M., Massini, G., Breda, M., Lodwick, W. A., Newman, F. and Asadi-Zeydabadi, M. (2018). Artificial Neural Networks, In *Artificial Adaptive Systems Using Auto Contractive Maps* (pp. 11–35). Springer, Cham.
- Chai, T. and Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)?—Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, 7(3), pp. 1247–1250.
- Gerstner, W. and Kistler, W. M. (2002). Mathematical formulations of Hebbian learning, *Biological cybernetics*, 87(5-6), pp. 404–415.
- Graves, A. B. (2016). U.S. Patent No. 9,263,036, Washington, DC: U.S. Patent and Trademark Office.
- Guoguang, He., Mansih, D. S. and Kazuyuki, A. (2008). Threshold control of chaotic neural network, *Neural Networks*, 21, No. 2-3, pp. 114–121.
- Hamadneh, N., Sathasivam, S., Tilahun, S.L. and Choon, O.H. (2012). Learning logic programming in radial basis function network via genetic algorithm, *Journal of Applied Sciences*, 12(9), pp. 840–847.

- Kasihmuddin, M. S. M., Mansor, M. and Sathasivam, S. (2017a). Robust artificial bee colony in the Hopfield network for 2-satisfiability problem, *Pertanika Journal of Science & Technology*, 25(2), pp. 453–468.
- Kasihmuddin, M. S. M., Mansor, M. A. and Sathasivam, S. (2017b). Hybrid genetic algorithm in the Hopfield network for logic satisfiability problem, *Pertanika Journal of Science & Technology*, 25(1), pp. 139–152.
- Kasihmuddin, M. S. M., Mansor, M. A. and Sathasivam, S. (2018). Discrete hopfield neural network in restricted maximum k-satisfiability logic programming, *Sains Malaysiana*, 47(6), pp. 1327–1335.
- Kasihmuddin, M. S. M., Mansor, M. A., Alzaeemi, S., Basir, M. F. M. and Sathasivam, S. (2019, November). Quality Solution of Logic Programming in Hopfield Neural Network. In *Journal of Physics: Conference Series* (Vol. 1366, No. 1, p. 012094), IOP Publishing.
- Kowalski, R.A. (1979). *The Logic for Problem Solving*, New York: Elsevier Science Publishing.
- Mansor, M. A. B., Kasihmuddin, M. S. B. M. and Sathasivam, S. (2017). Robust artificial immune system in the Hopfield network for maximum k-satisfiability, *International Journal of Interactive Multimedia and Artificial Intelligence (IJIMAI)*, 4(4), pp. 63–71.
- Peter, B. and Carsten, M. (2017). Computational complexity and human decision-making, *Trends in Cognitive Sciences*, 21(12), pp. 917–929.
- Sathasivam, S. (2010). Upgrading logic programming in the Hopfield network, *Sains Malaysiana*, 39(1), pp. 115–118.
- Sathasivam, S. (2012). Applying Different Learning Rules in Neuro-Symbolic Integration, In *Advanced Materials Research* (Vol. 433, pp. 716–720), Trans Tech Publications Ltd.
- Sathasivam, S. (2015). Acceleration technique for neuro symbolic integration, *Applied Mathematical Sciences*, 9(9), pp. 409–417.
- Sathasivam, S., Mansor, M., Kasihmuddin, M. S. M. and Abubakar, H. (2020). Election algorithm for random k satisfiability in the Hopfield neural network. *processes*, 8(5), pp. 568, <https://doi.org/10.3390/pr8050568>.
- Velavan, M., Yahya, Z.R., Abdul Halif, M.N. and Sathasivam, S. (2016). Mean field theory in doing logic programming using Hopfield network, *Modern Applied Science*, 10(1), pp. 154–160.
- Yolcu, E. and Poczos, B. (2019). Learning Local Search Heuristics for Boolean Satisfiability, In *Advances in Neural Information Processing Systems* (pp. 7990–8001).

Biographies



Hamza Abubakar received both his B.Sc (Mathematics) and MSc (Financial Mathematics) from the University of Abuja, Nigeria in (2006) and (2014) respectively. He is currently pursuing a PhD degree in Mathematical Science, Universiti Sains Malaysia. Hamza joint the service of Isa Kaita College of Education, Dutsin-ma, Katsina, Nigeria in 2008 and raised from an assistant lecturer to a Senior lecturer in Mathematics and computers. He is an active member of the Nigerian Mathematical Society, Mathematical Association of Nigeria, Science Teachers Association of Nigeria and International Association of Engineers (OR and AI). His research interest include, Financial Mathematics, neural network modelling and metaheuristics optimization.



Sagir Abdu Masanawa is a Senior Lecturer in the Department of Mathematical Sciences, Federal University Dutsin-ma, Nigeria. He received his BSc Mathematics, Executive PGD in Computer Studies, MSc Information Technology and MTech Mathematics from Bayero University Kano, A.T.B.U. Bauchi, National Open University of Nigeria and Federal University of Technology Minna respectively. He became a Doctor of Philosophy in Mathematics (Neural Network) from Universiti Sains Malaysia. His current research interest includes neural networks, Data mining, Metaheuristic algorithms, Information Technology and Numerical methods.



Surajo Yusuf obtained his BSc. (Mathematics) and PGD (Computer Science) both at Bayero University Kano, Nigeria. He also obtained PGD in Education at Federal College of Education Kano, Nigeria. Surajo received his MSc. In Mathematical Sciences from the Universiti Teknologi Malaysia. He is currently a Senior lecturer with the Department of Mathematics, Isa Kaita College of Education Dutsin-ma, Katsina state Nigeria. His research interest include, Topology and modelling of networks.